



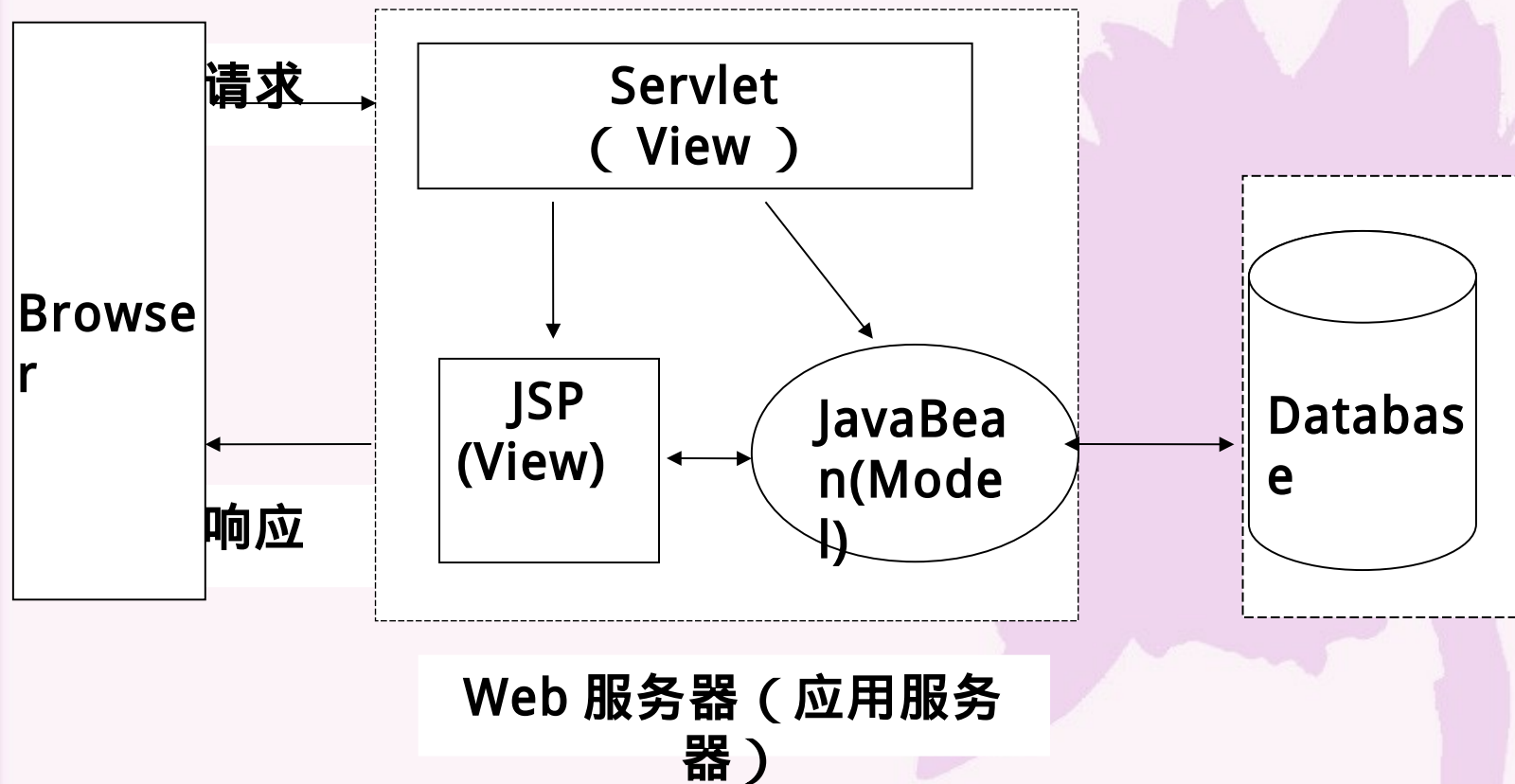
WEB

第 9 章 MVC 综合开发

本章目录

- 9.1 系统概述
- 9.2 系统概要设计
- 9.3 系统详细设计
- 9.4 本章小结

什么是 MVC 框架



9.1 系统概述

系统的设计思想：

1. 系统功能总体分为前台显示与后台维护两大部分

2. 按照 MVC 的开发框架进行设计：

**{ 模型层：JavaBean 以及 DAO 实现
视图层：JSP
控制层：Servlet**

本系统的特点：

页面模块化：

系统的界面在设计上都采用了模块化的设计思想，把多个页面共有的部分集成为一个模块。

业务处理封装化：

采用用户界面层 - 业务逻辑处理层 - 数据存储层三层结构设计。

面向对象设计：

采用面向对象的开发思想，将各个实体封装成对应的数据类，并为每个数据类开发一个操作类。

9.2 系统概要设计

9.2.1 系统需求分析

9.2.2 系统功能模块划分

9.2.3 数据库设计

9.2.1 系统需求分析

本外贸系统是一个中小型的外贸行业服务平台，功能主要是对各种信息的查询、添加、修改与删除。

由于本系统主要是针对这些信息而设计的，所以其功能主要集中在信息的查阅与操作方面。

9.2.1 系统需求分析

前台客户端部分

- (1) 首页界面设计
- (2) 产品信息查阅
- (3) 外贸信息查阅与搜索
- (4) 使用指南查阅
- (5) 客户服务查阅
- (6) 系统公告查阅

后台管理员部分

- (1) 管理员登陆管理
- (2) 公告信息管理
- (3) 外贸信息管理
- (4) 使用指南信息管理
- (5) 客户服务信息管理等

9.2.2 系统功能模块划分

后台管理

管理员登录

类别维护

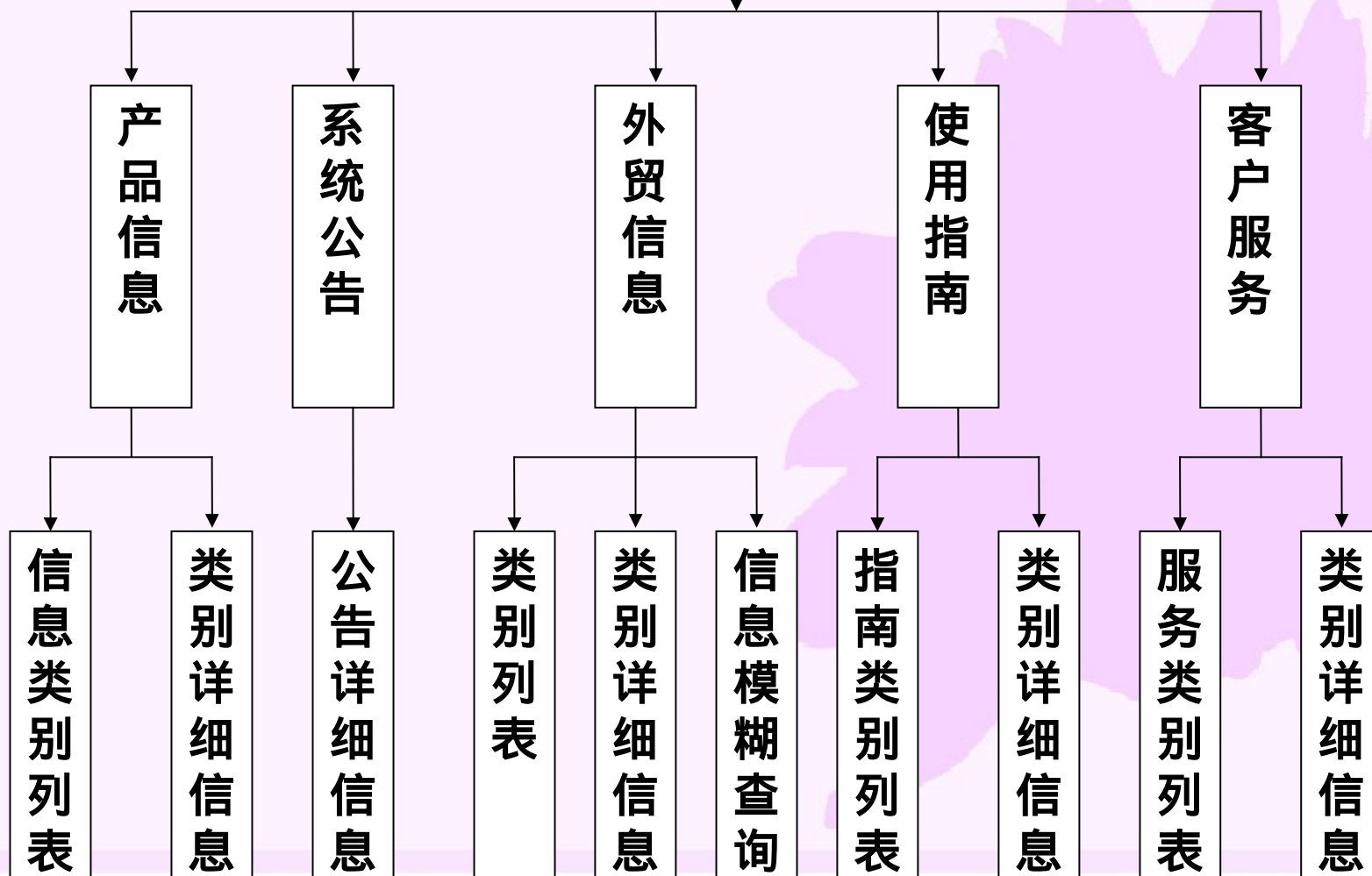
信息发布维护

网站模块维护

管理员维护

首页

前台系统



9.2.2 数据库设计

根据需求设计数据库

后台管理员对数据的需求：

- 管理员有权修改自身登录后台的密码。
- 管理员可以对各功能模块的详细列表进行增删改等操作。
- 管理员可以对各项数据进行审核等。

9.2.2 数据库设计

一般用户对数据的需求：

- 导航栏可以链接到每个功能模块的分类清单。
- 每个功能模块都可进行再分类，比如外贸集粹又包括外贸基础、进出口退税等。
- 用户可以浏览所有客户端界面信息。
- 用户可以实现对外贸集粹内所有信息的模糊查询。
- 进行流量统计，如在线人数、总访问量等。
- 提供管理员登录链接。
- 根据用户点击率对外贸集粹内所有信息进行排名推荐等。

9.2.2 数据库设计

主要的数据库项和数据结构

数据结构	数据项
管理员信息	用户 ID、用户名和密码
公告信息	公告 ID、公告标题和内容
外贸集粹信息	ID、标题、来源、日期、内容、点击率等
产品信息	ID、类型和内容
问题信息	ID、问题标题和解答等
服务信息	ID、服务类别和内容等

9.3 系统的详细设计

9.3.1 外贸集粹维护模块

一、模型层设计

二、控制层设计

三、视图层设计

9.3.1 外贸集粹维护模块

一、模型层设计

1、外贸集粹 javaBean (Guide.java)

```
public class Guide
{
    public Guide(){}
    private int tguideid;      // 编号
    private String tguidetitle; // 标题
    private String tguidefrom;  // 来源
    private Date tguidedate;    // 发布日期
    private String tguidecontent; // 正文
    private int tguidehits;     // 点击数
    private String tguidetype;  // 类型
    ... // 各属性的 getX ( ) 方法和 setXX() 方法参见源代码
```

9.3.1 外贸集粹维护模块

2. 分页显示 JavaBean (PageInfo.java)

PageInfo 的属性及两个构造函数：

```
private int pagesize = 10; // 页面大小
private int curpage;      // 当前页页码
private int prepage;      // 上一页页码
private int nxtpage;      // 下一页页码
private int totalpage;    // 总页码数
private int totalitem;    // 总条目数
private int from;         // 从 from 下一条开始取记录
private String next="";
private String previous="";
public PageInfo(int curpage) {
    this.curpage = curpage;
}
```


9.3.1 外贸集粹维护模块

```
public PageInfo(int curpage, int pagesize) {  
    this.curpage = curpage;  
    this.pagesize = pagesize;  
}  
// 此 bean 里的一个主要方法  
public void CaculatePageInfo(ResultSet rs)  
    throws SQLException {  
    // 1. 获取总条目数量  
    rs.last();  
    this.totalitem = rs.getRow();  
    // 2. 计算页码信息  
    this.totalpage = (totalitem % pagesize == 0) ?  
        (totalitem / pagesize) :  
        (totalitem / pagesize + 1);  
}
```

9.3.1 外贸集粹维护模块

// 当前页，前页，后页等

```
if (this.totalpage == 0) {  
    this.curpage = 0;  
    this.prepage = 0;  
    this.nxtpage = 0;  
} else {  
    this.prepage = (this.curpage <= 1) ? 1 : (this.curpage-1);  
    this.nxtpage = (this.curpage >= this.totalpage) ? this.curpage  
        : (this.curpage + 1);  
    this.from = (this.curpage - 1) * this.pagesize+1;  
    this.previous=(this.curpage <= 1) ? "" : " 上一页 ";  
    this.next=(this.curpage >= this.totalpage) ? "" : " 下一页 ";  
}
```

9.3.1 外贸集粹维护模块

3.DAO 调用分页显示组

件 利用分页显示组件，我们就可以在 DAO 业务逻辑处理中对查询记录的结果进行分页显示操作，并且在相应的 Servlet 文件中创建 PageInfo 实例。比如在 GuideFindServlet.java 中，先用 currentpage、pageSize 两个变量分别接收传过来的页码参数和页面大小，然后创建 PageInfo 实例，为了区分是否为第一次访问页面，所以把默认时页码设为 1；对于没有设置每页显示记录条数时，默认为 20，然后将 PageInfo 实例以及数据查询结果集 pslist 一起保存在 Session 内置对象中。

9.3.1 外贸集粹维护模块

GuideFindServlet.java 中的主要实现代码：

1. 得到页面传过来的参数（若没有则使用默认值）

```
int pageSize=request.getParameter("PageSize")==null?  
    20:Integer.parseInt(request.getParameter("PageS  
        ize"));  
int currentpage=request.getParameter("page")==null?  
    1:Integer.parseInt(request.getParameter("page"));
```

9.3.1 外贸集粹维护模块

2. 实例化一个 PageInfo 对象，并传入 GuideDAO

```
PageInfo PageInfo=new PageInfo(currentpage,pagesize);  
GuideDAO GuideDAO=new GuideDAO(PageInfo);  
.....
```

3. 将相应的结果存入 session

```
request.getSession().setAttribute("pslist",pslist);  
request.getSession().setAttribute("PageInfo",PageInfo);
```

9.3.1 外贸集粹维护模块

4. 外贸集粹 DAO (GuideDAO.java)

GuideDAO 实现了 DAO 接口，并组合了一个 PageInfo 实例

```
public class GuideDAO implements DAO{  
    private PageInfo PageInfo;  
    public GuideDAO() {  
    }  
    public GuideDAO(PageInfo PageInfo) {  
        this.PageInfo = PageInfo;  
    }  
    .....  
}
```

GuideDAO 包含的方法

返回类型	方法	功能
void	insert(Object bean)	添加数据
void	update(Object bean)	更改数据
void	delete(Object PKey)	删除数据
List	findAll()	整体查询，即查询相关表内所有记录
List	findExecutingSQL(String sql, Object[] sqlParams)	筛选查询，即查询相关表内符合一定条件的所有记录
void	populateVO(Guide bean, ResultSet rs)	将查询记录结果置入 Bean 中
List	fetchMultiResults(ResultSet rs)	将 Bean 添加到 List 对象中
void	edithit(Object PKey)	更新点击率
List	findHits()	按照点击率高低排序查询记录

9.3.1 外贸集粹维护模块

有关于前三个方法，都是调用 sql 语句进行相关数据库的操作，这里注意下面的方法实现：

```
public List findExecutingSQL(String sql, Object[] sqlParams) {  
    List pslist = null;  
    try {  
        DBConnect dbc = new DBConnect(sql,  
ResultSet.TYPE_SCROLL_SENSITIVE,    ResultSet.CONCUR_UPDATABLE);  
        for (int i = 0; i < sqlParams.length; i++) {  
            dbc.setObject(i + 1, sqlParams[i]);  
        }  
        ResultSet rs = dbc.executeQuery();  
        pslist = fetchMultiResults(rs);  
        dbc.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return pslist;  
}
```


9.3.1 外贸集粹维护模块

populateVO 方法是关于将记录集里的信息映射存储到 bean 的方法。

```
protected void populateVO(Guide bean, ResultSet rs) throws
    SQLException {
    bean.setTGuideID(rs.getInt("TGuideID"));
    bean.setTGuideContent(rs.getString("TGuideContent"));
    ;
    bean.setTGuideFrom(rs.getString("TGuideFrom"));
    bean.setTGuideTitle(rs.getString("TGuideTitle"));
    bean.setTGuideDate(rs.getDate("TGuideDate"));
    bean.setTGuideHits(rs.getInt("TGuideHits"));
    bean.setTGuideType(rs.getString("TGuideType"));
}
```

9.3.1 外贸集粹维护模块

fetchMultiResults(ResultSet rs) 方法则是在 populateVO 的基础上将记录集的所有信息存入一个 List<Guide> 中，其中还要考虑到分页信息。核心代码

```
rs.first();
if(PageInfo==null) {
    while (!rs.isAfterLast()) {
        Guide bean = new Guide();
        populateVO(bean,rs);
        resultList.add(bean);
        rs.next();
    }
} else{
```

9.3.1 外贸集粹维护模块

```
        rs.first();  
    }  
    PageInfo.CaculatePageInfo(rs);  
    rs.absolute(PageInfo.getFrom());    // 绝对定位  
    while ((!rs.isAfterLast()) && (CurrNum <  
        PageInfo.getPagesize())) {        Guide bean  
= new Guide();  
        populateVO(bean, rs);  
        resultList.add(bean);  
        rs.next();  
        CurrNum++;  
    }  
}
```

9.3.1 外贸集粹维护模块

edithit(Object PKey) 方法

edithit(Object PKey) 方法，实现信息点击率的更新，核心代码：

```
String sql = "Update TGuide set  
TGuideHits=TGuideHits+1 where TGuideID=?";  
DBConnect dbc = new DBConnect(sql);  
dbc.setObject(1, PKey);  
dbc.executeUpdate();
```

9.3.1 外贸集粹维护模块

findHits() 方法

findHits() 方法，实现信息点击率的查询

核心代码：

```
String sql = "SELECT * FROM TGuide order  
by TGuideHits desc";
```

```
DBConnect dbc = new DBConnect(sql,  
ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_UPDATABLE);
```

```
ResultSet rs = dbc.executeQuery();
```

```
pslist = fetchMultiResults(rs);
```

```
dbc.close();
```

9.3.1 外贸集粹维护模块

二、控制器层设计

类名	功能
GuideFindServlet	负责查询的的流程控制
GuideUpdateServlet	负责数据更新的的流程控制
GuideDeleteServlet	负责数据删除的的流程控制
GuideDetailServlet	负责数据添加与更改的的流程控制

GuideFindServlet 的查询流程

一. 先获取 currentpage、 pageSize 两个参数，然后创建 PageInfo 实例，如果得不到 currentpage，则默认为 1；如果得不到 pageSize，则默认为 20

```
int pageSize=request.getParameter("iPageSize")==null?20
    :Integer.parseInt(request.getParameter("iPageSize"));
int currentpage=request.getParameter("page")==null?1
    :Integer.parseInt(request.getParameter("page"));
PageInfo PageInfo=new PageInfo(currentpage,pageSize);
GuideDAO GuideDAO=new GuideDAO(PageInfo);
```

将 PageInfo 实例以及数据查询结果集 pslist 一起保存在 Session 内置对象中。

1、判断传入参数 Type 的值的的情况：

Type 为 null：

➤ 栏目类型为 null：

```
plist=GuideDAO.findAll();
```

➤ 栏目类型不为 null：

```
Object[] sqlparams=new Object[1];
```

```
sql="SELECT * FROM TGuide where TGuideType=? order  
by TGuideDate desc,TGuideID desc";
```

```
sqlparams[0]=stype;
```

```
pslist=GuideDAO.findExecutingSQL(sql, sqlparams);
```


Type 为 "search" :

➤ 搜索类型为 null :

```
sql = "SELECT * FROM TGuide where TGuideTitle  
like ? order by TGuideDate desc,TGuideID desc";  
Object[] sqlparams=new Object[1];  
sqlparams[0]="%" + keyword + "%";  
pslist=GuideDAO.findExecutingSQL(sql,sqlparams  
);
```

Type 为 "search" :

➤ 搜索类型不为 null :

```
sql = "SELECT * FROM TGuide where TGuideTitle  
like ? and TGuideType=? order by TGuideDate  
desc,TGuideID desc";
```

```
Object[] sqlparams=new Object[2];
```

```
sqlparams[0]="%" + keyword + "%";
```

```
sqlparams[1]=stype;
```

```
pslist=GuideDAO.findExecutingSQL(sql,sqlparams);
```

2、判断传入的 from 值的情况：
当传入的为" index" 时，调用 GuideDAO.findHits();
并将结果存入 pslist。

做完这些工作就将结果存入 session：

```
request.getSession().setAttribute("type",type);  
request.getSession().setAttribute("key",keyword);  
  
request.getSession().setAttribute("pslist",pslist);  
request.getSession().setAttribute("PageInfo",PageIn  
fo);
```

GuideFindServlet 的查询流程

二、控制流程

```
if (from!=null&&from.equals("index")){  
    if (type==null) {  
        response.sendRedirect("foreinguide.jsp");  
    }  
    else if (type.equals("search")){  
        response.sendRedirect("foreinguide.jsp?  
            type=search&keyword="+  
            java.net.URLEncoder.encode(keyword)+"");  
    }  
}
```

GuideFindServlet 的查询流程

```
response.sendRedirect("foreinguide.jsp?
type=search&keyword="+keyword+"");
    return;
}
} else {
    response.sendRedirect
        ("fadmin/foreigntradeguide.jsp");
    return;
}
```

GuideUpdateServlet

GuideUpdateServlet.java 的作用是编辑外贸集粹信息，根据传递的 type 参数值决定本次操作是增加还是修改一条记录，核心代码：

TGuideContent TGuideDate TGuideFrom TGuideTitle TGuideType
均为前段传过来的 String 类型参数

```
Guide bean=new Guide();
```

```
bean.setTGuideContent(TGuideContent);
```

```
bean.setTGuideDate(TGuideDate);
```

```
bean.setTGuideFrom(TGuideFrom);
```

```
if (type.equals("edit")){
```

```
    int guideid=Integer.parseInt
```

```
        (request.getParameter("guideid"));
```

```
    bean.setTGuideID(guideid);
```

```
}
```

GuideUpdateServlet

```
bean.setTGuideTitle(TGuideTitle);
bean.setTGuideType(TGuideType);
GuideDAO guideDAO=new GuideDAO();
try{
    if (type.equals("edit"))
        guideDAO.update(bean);
    else
        guideDAO.insert(bean);
    response.sendRedirect("GuideListServlet");
}
catch(Exception e)
{
    response.sendRedirect(Const.errorpage);
}
```

GuideDeleteServlet

根据传递的主键值删除对应的记录

```
GuideDAO guideop=new GuideDAO();
    int guideid=Integer.valueOf
        (request.getParameter("guideid"));
    try{
        guideop.delete(guideid);
        response.sendRedirect("GuideListServlet");
    }
    catch(Exception e) {
        response.sendRedirect(Const.errorpage);
    }
}
```


GuideDetailServlet

查询到所需的外贸集粹信息对象，存储到 session 并重定向到 JSP 页面
核心代码：

```
// 获取页面参数：
int guideid=Integer.parseInt(request.getParameter("guideid"));
String from=StrFun.chinese(request.getParameter("from"));
String type=StrFun.chinese(request.getParameter("type"));
// 定义并初始化相关对象
GuideDAO GuideDAO=new GuideDAO();
Object[] sqlparams=new Object[1];
sqlparams[0]=guideid;
```

GuideDetailServlet

```
if (sType!=null) {
    sql="select * from TGuide where TGuideID=? and
        TGuideType='"+ sType+"'";
} else{
    sql="select * from TGuide where TGuideID=? ";
}
pslist=GuideDAO.findExecutingSQL(sql, sqlparams);
if(!pslist.isEmpty()){
    Iterator it=pslist.iterator();
    Guide guideobject=(Guide)it.next();
    request.getSession().setAttribute("guideobject",
guideobject);
}
```

GuideDetailServlet

```
if (from!=null&&from.equals("index")){  
    GuideDAO.edithit(guideid);  
    response.sendRedirect("display.jsp");  
    return;  
}  
else{  
  
    response.sendRedirect("fadmin/foreigntradeguidedo.js  
        p?type="+type+"");  
    return;  
}
```


以上页面采用框架式网页的模式进行设计，框架分成左右两部分，分别链接到 left.jsp 和 main.jsp。以下是管理页面 fadmin/productinfo.jsp 主要代码：

```
<script language="javascript">
function del() {
if (confirm(" 确定删除吗？ ")==true)
    return true;
else
    return false;
}
function GoToPage() {
    ipageno=document.all.Sltpageno.value;
    document.location.href =
        "../GuideListServlet?iPageNo=" + ipageno+
        "";
}
```

9.3.1 外贸集粹维护模块

页面元素核心代码：

```
<TR><TD class=tdbg3 height=5></TD></TR>
<TR><TD class=tdbg1 vAlign=top align="center">
<% if (request.getSession().getAttribute("pslist")!=null){
ArrayList pslist=(ArrayList)session.getAttribute("pslist");
Iterator it= pslist.iterator();
    %>
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr align="center">
    <td width="30%"> 标题 </td>
    <td width="15%"> 类型 </td>
    <td width="22%"> 发布日期 </td>
    <td width="10%"> 点击次数 </td>
    <td colspan="2"> 管理 [
<a href=" ../GuideDetailServlet?guideid=0&type=add"> 添加 </a>]</td>
</tr>
```

9.3.1 外贸集粹维护模块

```
<% while(it.hasNext()){  
    Guide guide=(Guide)it.next();  
%>  
<tr align="center">  
    <td><%=StringDeal.cutstring(guide.getTGuideTitle(),15) %></td>  
    <td><%=guide.getTGuideType() %></td>  
    <td><%=guide.getTGuideDate() %></td>  
    <td><%=guide.getTGuideHits() %></td>  
    <td width="15%">  
[<a href="../GuideDetailServlet?guideid=<%=guide.getTGuideID()  
%>&type=edit"> 编辑 </a>]</td>  
    <td width="15%">  
[<a href="../GuideDelServlet?guideid=<%=guide.getTGuideID() %>"  
onClick="return del();"> 删除 </a>]</td>  
</tr>  
<%}%>
```

9.3.1 外贸集粹维护模块

```
<tr align="center">
<td colspan="6"><%if(session.getAttribute("PageInfo")!=null){
    PageInfo info=(PageInfo)session.getAttribute("PageInfo");
    %>
    <a href=" ../GuideListServlet?page=<%=info.getPrepage()%>">
    <%=info.getPrevious()%></a>
    <a href=" ../GuideListServlet?page=<%=info.getNxtpage()%>">
    <%=info.getNext()%></a>
    <% } %>
</td>
</tr>
</table>
<%} %>
```


9.3.1 外贸集粹维护模块

以下是信息添加页面

→ 欢迎进入控制面板

外贸集粹管理

功能菜单

管理首页 退出

首页管理

消息管理

产品信息

外贸集粹

使用指南

客户服务

关于我们

系统登陆管理

标题:

出处:

类型: 外贸基础

更新时间: 2008-03-02

段落样式 字体 字号 100% B I U

主要内容:

代码 设计 预览

提交 重填 返回

外贸集粹的信息添加页
Web 程序设计 第九章

页面的 js 代码 (主要是对 text 输入框的是否为空的验证) :

```
<script language="javascript">
function checkform(){
    TGuideTitle=document.all.TGuideTitle.value;
    TGuideFrom=document.all.TGuideFrom.value;
    if (TGuideTitle==""){
        alert(" 请填写标题 ");
        return false;
    }else if(TGuideFrom==""){
        alert(" 请填写出处 ");
        return false;
    }else if(eWebEditor1.getHTML()==""){
        alert(" 请填写内容 ");
        return false;
    }else
        return true;
}
</script>
```

页面端元素代码：

```
<%
```

```
Calendar cal = Calendar.getInstance();
```

```
SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
```

```
String nowdate = sf.format(cal.getTime());
```

```
// 以上两句代码获得当前日期
```

```
String
```

```
type=StrFun.chinese(request.getParameter("type"));
```

```
// 获得请求中的 type 属性，判断是否是
```

```
edit，//StrFun.chinese() 为处理中文字符串的函数避免出现中 // 文乱码
```

```
%>
```

页面中的元素：

- ◆标题 :<input type='text' name="TGuideTitle" >
- ◆出处 :<input type='text' name="TGuideFrom" >
- ◆类型 :<select name="TGuideType">
 - <option value=" 外贸基础" selected> 外贸基础</option>
 - <option value=" 政策法规 "> 政策法规 </option>
 - <option value=" 进出口退税 "> 进出口退税 </option>
- ◆更新日期 :<input type='text' name="TGuideDate">
- ◆主要内容 :<textarea type='text' name="TGuideContent">
- ◆提交与重置按钮就不在过多阐述

9.3.1 外贸集粹维护模块

当 request 的请求中 type 为 edit 时：

以上元素的默认值 value 将从 session 中取得，例如标题的代码为

```
value="<%
```

```
if (type.equals("edit")){
```

```
Guide
```

```
guideobject=(Guide)request.getSession().getAttribute("guideobject");
```

```
out.print(guideobject.getTGuideTitle());
```

```
} %>">
```

其他的元素类似。

此页面中还存在一个特殊的元素：

```
<IFRAME ID="eWebEditor1" src="../eWebEditor/eWebEditor.jsp?  
    id=TGuideContent&style=standard" frameborder="0"  
    scrolling="no" width="650" height="350">  
</IFRAME>
```

采用了在线编辑器（ eWebEditor ）组件，该组件使用方便、灵巧，可以满足多种网页资源的编辑需求。其中 <IFRAME> 中的 `src="../eWebEditor/eWebEditor.jsp?id=TguideContent&....."` 中的 `id=TGuideContent` 必须和文本域的名字（即 `name="TGuideContent"` ）相同即可。若要了解详细内容可以再网上查阅相关资料。

该页面详细代码参见 `fadmin/foreigntradeguidedo.jsp`

9.3.2 管理员维护模块

一、模型层设计

(1) 管理员 JavaBean

```
public class Tadmin{  
    private Integer tadminId;  
    private String tadminUsername;  
    private String tadminPass;  
    .....    // 各属性的 getX ( ) 方法和  
               setXX ( ) 方法略
```

9.3.2 管理员维护模块

(2) 管理员 DAO(与 GuideDAO 相似)

返回类型	方法	功能
void	insert(Object bean)	添加数据
void	update(Object bean)	更改数据
void	delete(Object PKey)	删除数据
List	findAll()	整体查询，即查询相关表内所有记录
List	findExecutingSQL(String sql, Object[] sqlParams)	筛选查询，即查询相关表内符合一定条件的所有记录

9.3.2 管理员维护模块

二、控制层设计：

类名	功能
LoginServlet	负责登录控制
LogoutServlet	退出控制
AdminUpdateServlet	负责更新管理员信息的 的流程控制

管理员维护模块包含的
Servlet

LoginServlet.java

核心代码：

```
String rand=request.getSession().getAttribute("rand").toString();
String userpwd=request.getParameter("userpwd");
String input =request.getParameter("rand");
String username=request.getParameter("username");
String url="";
if (!rand.equals(input)){
    response.sendRedirect("fadmin/adminlo.jsp");
} else{
    TAdminDAO adminop=new TAdminDAO();
    String sql="select * from TAdmin where
TAdminUsername=?
        and TAdminPass=?";
    Object[] sqlparams=new Object[2];
```

LoginServlet.java

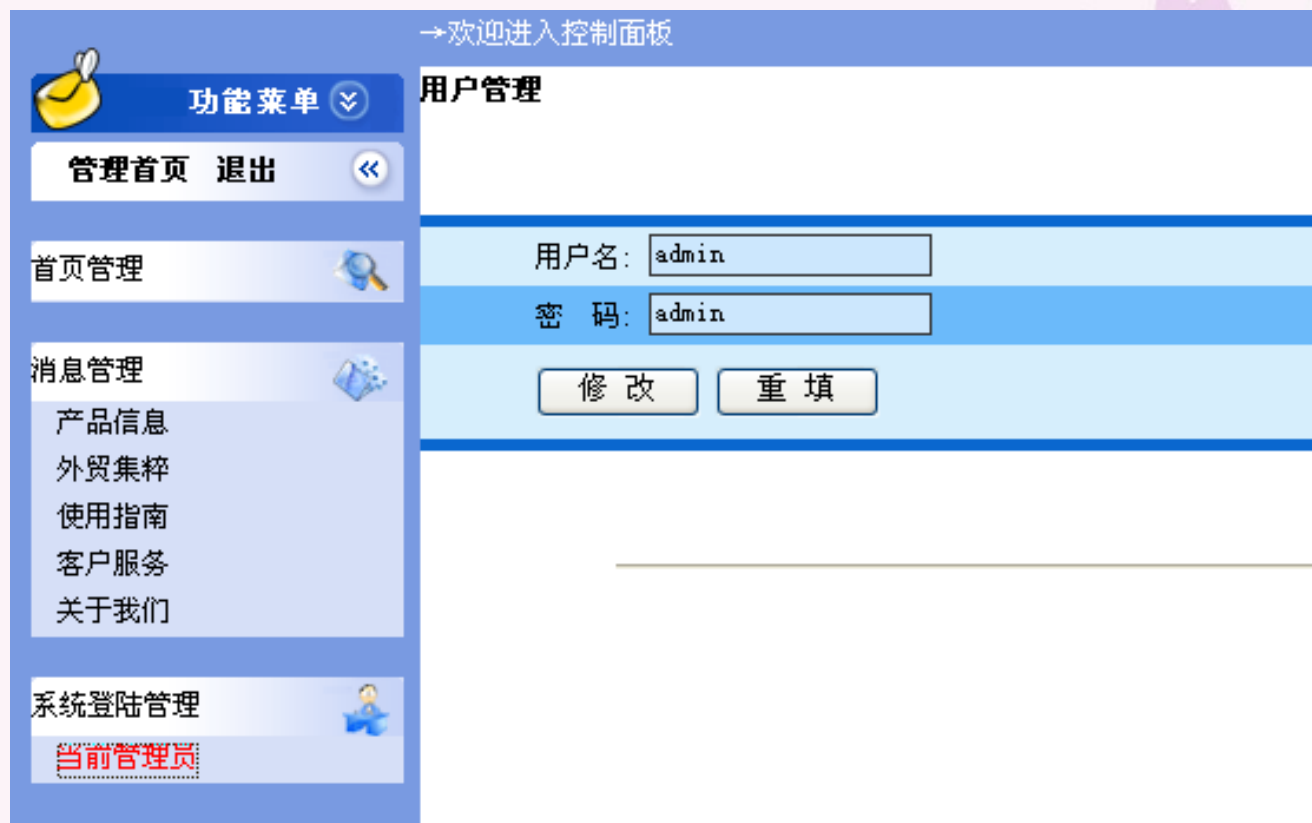
```
sqlparams[0]=username;  
sqlparams[1]=userpwd;  
List pslist=null;  
Tadmin admin=null;  
pslist=adminop.findExecutingSQL(sql,sqlparams);  
Iterator it=pslist.iterator();  
while(it.hasNext()){  
    admin=(Tadmin)it.next();  
}
```

LoginServlet.java

```
if (admin!=null) {  
    request.getSession().setAttribute("admin","admin");  
    request.getSession().setAttribute("username",username);  
    request.getSession().setAttribute("userpwd",userpwd);  
    request.getSession().  
        setAttribute("id",admin.getTadminId());  
        url="fadmin/index.jsp";  
} else {  
    url="fadmin/adminlo.jsp";  
    response.sendRedirect(url);  
}  
}
```

9.3.2 管理员维护模块

三、视图层设计



管理员管理页面效果图

9.3.2 管理员维护模块

主要代码如下：

```
<%
String userpwd=request.getSession().
    getAttribute("userpwd").toString();
String username=request.getSession().
    getAttribute("username").toString();
%>
.....
<form action="../EditAdminServlet" method="post">
    <TABLE height="100%" cellSpacing=0 cellPadding=0 width="100%"
align=center border=0>
<TBODY>
<%
if (request.getSession().getAttribute("id")!=null){
    int id=Integer.parseInt((request.getSession().
        getAttribute("id")).toString());
%>
```

```
<TR><TD class=tdbg3 height=5></TD></TR>
```

9.3.2 管理员维护模块

```
<TR><TD class=tdbg1 vAlign=top>
```

```
.....
```

```
<table width="100%" border="0" cellspacing="0"
cellpadding="0">
```

```
<tr>
```

```
    <td width="6%" height="25" class="tdbg1">&nbsp;  </td>
```

```
    <td width="88%" class="tdbg1" > 用户名 :
```

```
        <input name=username type="text" class="dataInput"
id="username"                size="30" value="<%=username
%>">&nbsp;  &nbsp; </td>
```

```
    <td width="6%" class="tdbg1" >&nbsp;  </td>
```

```
</tr>
```

```
<tr><td height="25" class="tdbg2">&nbsp;  </td>
```

```
    <td height="25" class="tdbg2"> 密 码 :
```

9.3.2 管理员维护模块

```
<input name="userpwd" type="text"  
      class="dataInput" id="userpwd" size="30"  
      value="<%=userpwd %>">&nbsp;&nbsp;   
      <font color="#FF0000">  
        <input name="TAdminID" type="hidden"  
          id="TAdminID" value="<%=id%>">  
      </font> </td>  
    <td height="25" class="tdbg2">&nbsp;</td>  
  </tr>  
  <tr><td height="40" class="tdbg1" >&nbsp;</td>  
  <td height="40" class="tdbg1" >  
    <input type="submit" name="btnSubmit" value=" 修改 ">  
    <input type="reset" name="btnReset" value=" 重填 "></td>
```

官埋员官埋奥面效果图

• • • • •

9.3.3 其它功能模块

一、模型层设计

(1)JavaBean:

类名	含义
Tannouce	公告信息
Tproduct	产品信息
Tquestion	问题信息
TService	服务信息

(2)DAO:

类名	功能
TannouceDAO	封装对公告信息的增、删、改、查询
TproductDAO	封装对产品信息的增、删、改、查询
TquestionDAO	封装对问题信息的增、删、改、查询
TServiceDAO	封装对服务信息的增、删、改、查询

以上 DAO 类均要实现 DAO 接口，因而必须首先实现 DAO 的所有增删改以及查询方法，具体实现方式与外贸集粹 DAO 类似

9.3.3 其它功能模块

二、控制器层设计

类名	功能
AnnouceFindServlet	负责公告信息查询的流程控制
AnnouceUpdateServlet	负责公告信息更新的流程控制
findServlet	负责产品信息查询的流程控制
updateServlet	负责产品信息更新的流程控制
QuesFindServlet	负责问题信息查询的流程控制
QuesUpdateServlet	负责问题信息更新的流程控制
QuesDeleteServlet	负责问题信息删除的流程控制
QuesDetailServlet	负责问题信息添加的流程控制
TServiceFindServlet	负责服务信息查询的流程控制
TServiceUpdateServlet	负责服务信息更新的流程控制

以上 Servlet 类的具体实现代码也与外贸集粹模块的 Servlet 类似

9.3.3 其它功能模块

三、视图层设计

系统中其它模块所使用的 JSP 文件数量较多，这里不再一一列出。作为 MVC 中的视图层，JSP 的作用无外乎两种：一种是包含各种输入控件的表单页，另一种则是负责把处理的结果呈现给客户端用户。

9.4 本章小结

本章详细论述了如何采用 MVC 框架来设计和实现外贸信息服务网站这样的典型的 Web 信息系统，并从实际开发的角度具体阐述了 MVC 设计思想的思想。本章内容着重对系统进行了总体功能模块划分、数据库设计和各功能模块详细设计，其中对 DAO 模式以及分页显示均做了详细的介绍，并用相关代码加以说明。从本系统的开发中，我们可以看出，MVC 开发框架通过强制性分离，使得 M、V、C 三个层次之间相互独立，实现了松耦合的设计要求，这样可以使得系统在健壮性、代码重用和代码结构化等方面更上一个新台阶。

本章习题

- 1、请简述 MVC 开发框架的思想以及各层面所包含的 Web 组件。
- 2、DAO 对象在本系统中是如何被调用的？
- 3、请简述分页显示功能是如何实现的？涉及哪些程序？
- 4、本系统前台的首页功能模块在实现过程中有何特殊之处？需要做怎样的技术处理？
- 5、请编写一个基于 MVC 的用户登录模块，要求如下：
 - （1）将检验用户名和密码的程序封装成模型层 DAO 类中的方法；
 - （2）用户登录后将其登录时间和次数更新到数据库里；
 - （3）用户登录后将其个人基本信息（JavaBean）存到 session 中，并在最终页面打印显示。